技术面试

一、必备知识

1·1、核心数据结构

数据结构	算　法	概　念
链表	广度优先搜索	位操作
树、单词查找树、图	深度优先搜索	内存（堆和栈）
栈和队列	二分查找	递归
堆	归并排序	动态规划
向量/数组列表	快排	大 O 时间及空间
散列表		

1.2、2的幂

2 的幂	准确值（X）	近　似　值	X 字节转换成 MB、GB 等
7	128		
8	256		
10	1024	一千	1 K
16	65 536		64 K
20	1 048 576	一百万	1 MB
30	1 073 741 824	十亿	1 GB
32	4 294 967 296		4 GB
40	1 099 511 627 776	一万亿	1 TB

二、解题步骤

问题解决流程图

1 听

仔细聆听问题描述。每一个细节都可能在优化算法时派上用场。

2 举例

例子一般要袖珍一些或特殊一点儿。仔细调试，想一想还有其他特殊情况吗？例子能覆盖所有情况吗？

3 蛮力法

先尽快想出一个蛮力法来解决问题。在此之前，不要试图开发出一个高效的算法。给出一个朴素的算法和其运行时间，然后在此基础上优化该算法。当然了，现在不要写代码！

BUD优化

B：瓶颈（bottleneck）

U：无用功（unnecessary work）

D：重复性工作（duplicated work）

4 优化

用BUD法优化你的朴素算法，也可以尝试以下方法。

- 寻找未利用的信息。一般你需要一个问题中的所有信息。
- 手动解决一个例题，然后逆向思考。你是怎么解决的？
- 给出不正确的解法，思考为什么失败。你能修复这类问题吗？
- 权衡时间与空间。这时散列表至关重要。

5 梳理

有了一个最优算法后，详细地回顾一遍你的算法，以确保写代码之前理顺每个细节。

6 实现

你的目标是写出一手漂亮的代码。从一开始就追求模块化，并且通过重构清理掉不漂亮的代码。

持续交流。你的面试官乐于了解你是如何解决问题的。

7 测试

请按以下顺序测试。

(1) 概念测试。像代码复查一样，仔细审查一遍代码。

(2) 异常或不标准的代码。

(3) 热点代码，比如计算节点和空节点。

(4) 小测试用例，比大的快且同样有效。

(5) 特殊或边缘情况。

当发现错误时，请小心修复。